Java_8_features

1.Word_Counter

Word_Counter.java

Java_8/src/java_8_features_assignment/Word_Counter.java

```java
import java.util.*;

public class Word_Counter {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Alice", "Enemies", "Charlie", "Jonathan", "Bob");
        long count = names.stream()
            .filter(name -> name.length() > 5)
            .count();
        System.out.println("Names longer than 5 chars: " + count);
    }
}

```

Console Progress

<terminated> Word_Counter [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x8

```
Names longer than 5 chars: 3
```

2.Simple_Name_Sorter

Simple_name_sorter.java

```java
package java_8_features_assignment;

import java.util.*;

public class Simple_name_sorter  {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Alice", "Bob marlie", "Even", "Charlie");
        names.sort((a, b) -> a.compareTo(b));
        System.out.println(names);
    }
}


```

Console Progress

<terminated> Simple_name_sorter [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36

```
[Alice, Bob marlie, Charlie, Even]
```

3.Safe_Divider

Safe_Divider.java

```java
package java_8_features_assignment;


import java.util.Optional;

public class Safe_Divider {
    public static Optional<Double> divide(int a, int b) {
        if (b == 0) {
            return Optional.empty();
        }
        return Optional.of((double)a / b);
    }

    public static void main(String[] args) {
        System.out.println(divide(12, 2).orElse(Double.NaN));
        System.out.println(divide(10, 0).orElse(Double.NaN));
    }
}
```

Console | Progress

```
<terminated> Safe_Divider [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-
6.0
NaN
```

4.Pretty_Jointer

Pretty_Joiner.java

```java
package java_8_features_assignment;

import java.util.*;
import java.util.stream.*;

public class Pretty_Joiner {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Alice", "Bob", "Carol");
        String result = names.stream()
            .collect(Collectors.joining(", "));
        System.out.println(result);
    }
}
```

Console | Progress

```
<terminated> Pretty_Joiner [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-wi
Alice, Bob, Carol
```

5.Name_Filter

Name_Filter.java

```java
package java_8_features_assignment;

import java.util.*;
import java.util.stream.*;

public class Name_Filter {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Alice", "Even", "Adam", "Boby");
        List<String> filtered = names.stream()
            .filter(name -> name.startsWith("A"))
            .collect(Collectors.toList());
        System.out.println(filtered);
    }
}
```

Console | Progress

```
<terminated> Name_Filter [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEAS
[Alice, Adam]
```

6.Sorter

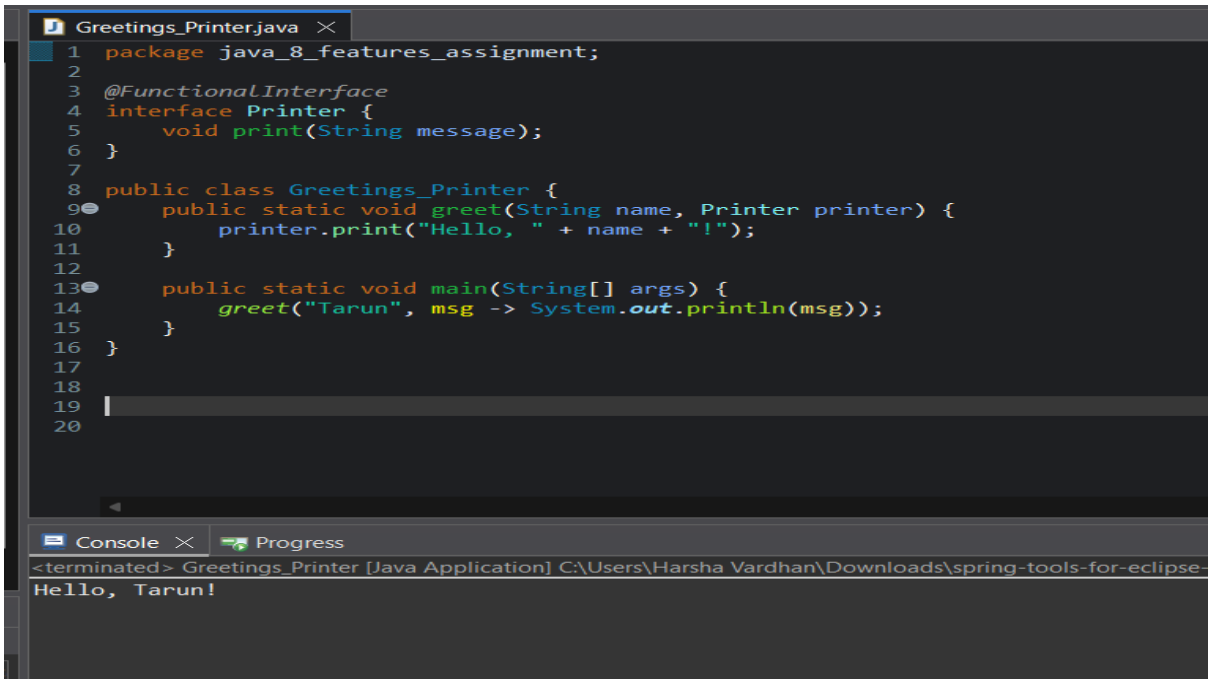
Method_Reference_Sorter.java

```java
package java_8_features_assignment;

import java.util.*;

public class Method_Reference_Sorter {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Charlie", "Alice", "Bob");
        names.sort(String::compareTo);
        System.out.println(names);
    }
}
```

Console | Progress

```
<terminated> Method_Reference_Sorter [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4
[Alice, Bob, Charlie]
```

7.Greetings_Printer



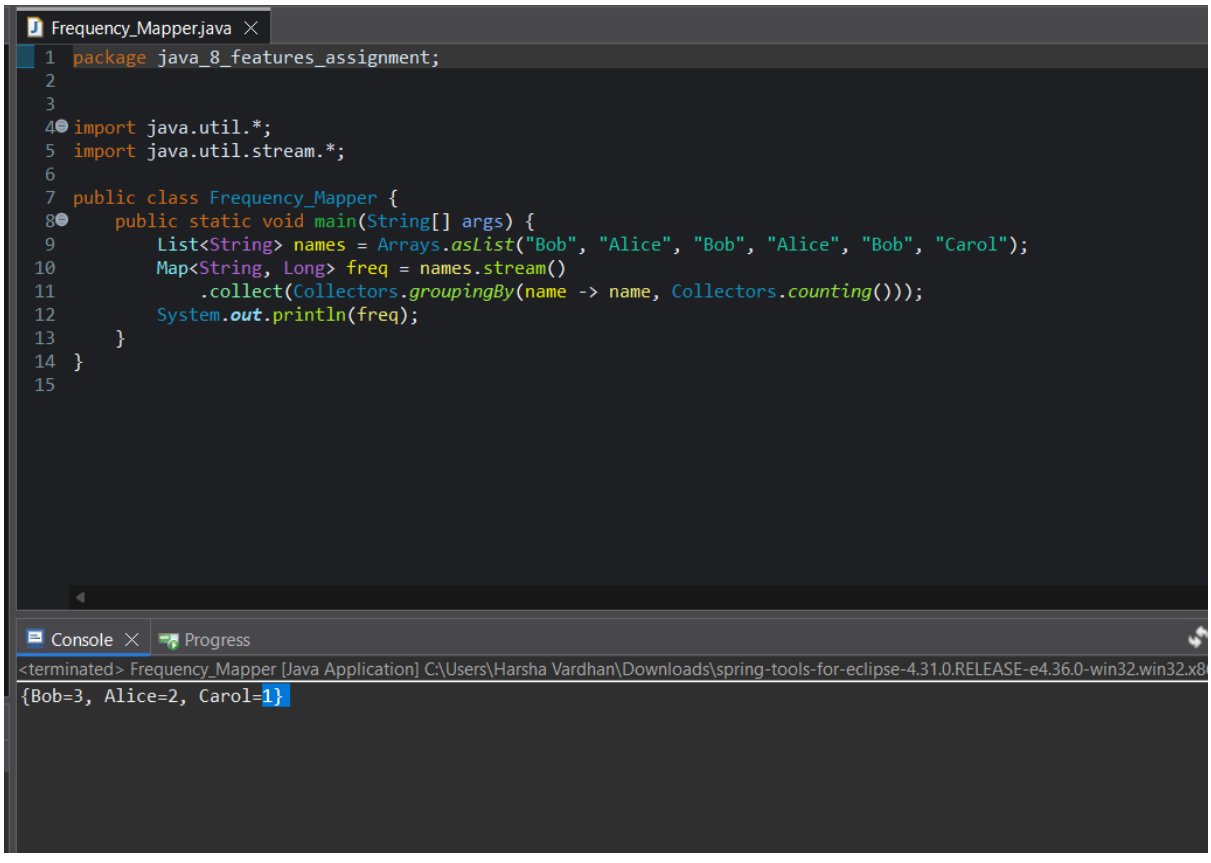
Greetings_Printer.java

```
1  package java_8_features_assignment;
2
3  @FunctionalInterface
4  interface Printer {
5      void print(String message);
6  }
7
8  public class Greetings_Printer {
9      public static void greet(String name, Printer printer) {
10         printer.print("Hello, " + name + "!");
11     }
12
13     public static void main(String[] args) {
14         greet("Tarun", msg -> System.out.println(msg));
15     }
16 }
17
18
19
20
```

Console | Progress

```
<terminated> Greetings_Printer [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-
Hello, Tarun!
```

8.Frequency _Mapper



Frequency_Mapper.java

```
1  package java_8_features_assignment;
2
3
4  import java.util.*;
5  import java.util.stream.*;
6
7  public class Frequency_Mapper {
8      public static void main(String[] args) {
9          List<String> names = Arrays.asList("Bob", "Alice", "Bob", "Alice", "Bob", "Carol");
10         Map<String, Long> freq = names.stream()
11             .collect(Collectors.groupingBy(name -> name, Collectors.counting()));
12         System.out.println(freq);
13     }
14 }
15
```

Console | Progress

```
<terminated> Frequency_Mapper [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-win32.win32.x8
{Bob=3, Alice=2, Carol=1}
```

9.Even_num_Collector

Even_no_collector.java

```java
package java_8_features_assignment;

import java.util.*;
import java.util.stream.*;

public class Even_no_collector {
    public static void main(String[] args) {
        List<Integer> numbers = Arrays.asList(1,2,3,4,5,6,7,8,9,10);
        List<Integer> evens = numbers.stream()
            .filter(n -> n % 2 == 0)
            .collect(Collectors.toList());
        System.out.println(evens);
    }
}
```

Console | Progress

```
<terminated> Even_no_collector [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-ecl
[2, 4, 6, 8, 10]
```

10.Default_Power

Default_Power.java

```java
package java_8_features_assignment;


interface Power {
    void showPower(int n);
    default void showDefault() {
        System.out.println("Default Power is: 42");
    }
}

public class Default_Power implements Power {
    public void showPower(int n) {
        System.out.println("Power is: " + n);
    }
    public static void main(String[] args) {
        Default_Power obj = new Default_Power();
        obj.showPower(10);
        obj.showDefault();
    }
}
```

Console | Progress

```
<terminated> Default_Power [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31
Power is: 10
Default Power is: 42
```

11.Custom_Filter_Chain

Custom_Filter_Chain.java

```java
package java_8_features_assignment;

import java.util.*;
import java.util.function.*;
import java.util.stream.*;

public class Custom_Filter_Chain {
    public static void main(String[] args) {
        List<String> names = Arrays.asList("Amy", "Alex", "Bob", "Anna", "Arthur");
        Predicate<String> startsWithA = name -> name.startsWith("A");
        Predicate<String> lengthIs4 = name -> name.length() == 4;
        List<String> filtered = names.stream()
            .filter(startsWithA.and(lengthIs4))
            .collect(Collectors.toList());
        System.out.println(filtered);
    }
}
```

Console | Progress

```
<terminated> Custom_Filter_Chain [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.RELEASE-e4.36.0-v
[Alex, Anna]
```

12.Parallel_Square_printer

Parallel_Square_Printer.java

```java
package java_8_features_assignment;

import java.util.*;

public class Parallel_Square_Printer {
    public static void main(String[] args) {
        List<Integer> nums = Arrays.asList(1,2,3,4,5,6);
        nums.parallelStream()
            .forEach(n -> System.out.println(n + "^2 = " + (n*n)));
    }
}
```

Console | Progress

```
<terminated> Parallel_Square_Printer [Java Application] C:\Users\Harsha Vardhan\Downloads\spring-tools-for-eclipse-4.31.0.
4^2 = 16
5^2 = 25
1^2 = 1
2^2 = 4
3^2 = 9
6^2 = 36
```